

# The Monte Carlo Method

Andreas Klappenecker

Texas A&M University

© 2018 by Andreas Klappenecker. All rights reserved.

We can try to estimate the value of  $\pi$  in the following way.

Choose a point  $(X, Y)$  uniformly at random in the  $2 \times 2$  square centered at  $(0, 0)$ . So  $X$  and  $Y$  are uniformly distributed random variables on the interval  $[-1, 1]$ .

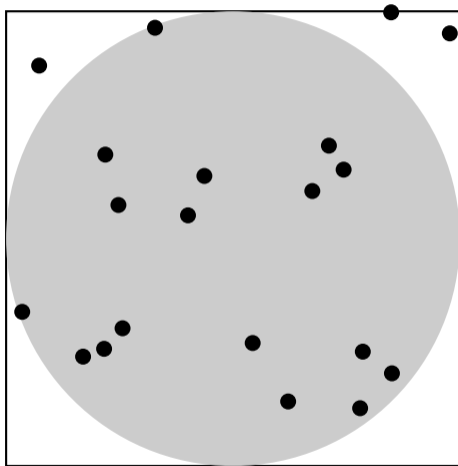
Define an indicator random variable  $Z$  for being in the unit circle by

$$Z = \begin{cases} 1 & \text{if } \sqrt{X^2 + Y^2} \leq 1, \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$\Pr[Z = 1] = \frac{\text{area of unit circle}}{\text{area of } 2 \times 2 \text{ square}} = \frac{\pi}{4}.$$

# Monte Carlo Approach for Estimating $\pi$



If we run this experiment  $m$  times with independently chosen coordinates, and  $Z_k$  is the value of the  $k$ -th run, then we expect for  $W = \sum Z_k$  to average

$$E[W] = E\left[\sum_{k=1}^m Z_k\right] = \sum_{k=1}^m E[Z_k] = \frac{m\pi}{4}.$$

Then  $W' = (4/m)W$  is a natural estimate for  $\pi$ .

By the Chernoff bound, the relative error is given by

$$\begin{aligned}\Pr[|W' - \pi| \geq \epsilon\pi] &= \Pr\left[\left|W - \frac{m\pi}{4}\right| \geq \frac{\epsilon m\pi}{4}\right] \\ &= \Pr[|W - \mathbb{E}[W]| \geq \epsilon\mathbb{E}[W]] \\ &\leq 2e^{-m\pi\epsilon^2/12}.\end{aligned}$$

Thus, if we use sufficiently many repetitions  $m$ , then we get an approximation to  $\pi$  that is as tight as we wish.

## Definition

A randomized algorithm gives an  $(\epsilon, \delta)$ -approximation for a value  $V$  if the output  $X$  of the algorithm satisfies

$$\Pr[|X - V| \leq \epsilon V] \geq 1 - \delta.$$

## Example

Our randomized algorithm for estimating  $\pi$  provides an  $(\epsilon, \delta)$ -algorithm if we choose

$$m \geq \frac{12 \ln(2/\delta)}{\pi \epsilon^2}$$

then

$$\Pr[|W' - \pi| \geq \epsilon\pi] \leq 2e^{-m\pi\epsilon^2/12} \leq \delta$$

or

$$\Pr[|W' - \pi| \leq \epsilon\pi] \geq 1 - 2e^{-m\pi\epsilon^2/12} \geq 1 - \delta.$$

## Proposition

Let  $X_1, \dots, X_m$  be independent and identically distributed indicator random variables, with  $\mu = \mathbb{E}[X_k]$ . If  $m \geq \frac{3 \ln(2/\delta)}{\epsilon^2 \mu}$ , then

$$\Pr \left[ \left| \frac{1}{m} \sum_{k=1}^m X_k - \mu \right| \geq \epsilon \mu \right] \leq \delta.$$

In other words,  $m$  samples provide an  $(\epsilon, \delta)$ -approximation for  $\mu$ .



# Fully Polynomial Randomized Approximation Schemes

## Idea

In general, we want an algorithm that approximates not just a single value but instead takes as input a problem instance and approximates the solution value for that problem. Here we are considering problems that map inputs  $x$  to values  $V(x)$ .

## Example

Given an input graph, we might want to know an approximation to the number of independent sets in the graph.

## Definition

A **fully polynomial randomized approximation scheme** (or shortly FPRAS) for a problem is a randomized algorithm for which, given an input  $x$  and any parameters  $\epsilon$  and  $\delta$  with  $0 < \epsilon, \delta < 1$ , the algorithm outputs an

$(\epsilon, \delta)$ -approximation to  $V(x)$

in a time that is polynomial in  $1/\epsilon$ ,  $\ln(1/\delta)$ , and the size of the input  $x$ .

# The DNF Counting Problem

### Problem

*Suppose that you are given a formula  $f$  in disjunctive normal form, that is,  $f$  is a disjunction of clauses that consist of conjunctions of literals.*

*Count the number of satisfying assignments to  $f$ .*

## Example

Consider the following example of a boolean function in DNF:

$$f(x_1, x_2, x_3, x_4) = (x_1 \wedge \bar{x}_2 \wedge x_3) \vee (x_2 \wedge x_4) \vee (\bar{x}_1 \wedge x_3 \wedge x_4).$$

Finding a satisfying assignment is always easy in DNF: Choose a clause  $C$  and assign truth values such that each literal in  $C$  evaluate to true. For example, if we choose

$$v(x_1) = T, v(x_2) = F, v(x_3) = T, v(x_4) = T/F,$$

then the first clause evaluates to true and hence  $v(f) = T$ .

## Example

Counting the number of satisfying assignment of a boolean function in DNF is not easy.

Indeed, if it were, then we could solve any instance  $g$  of SAT in  $n$  variables. Indeed, we could negate  $g$  use de Morgan's laws to obtain a DNF formula. Then  $g$  is satisfiable if and only if  $\bar{g}$  has less than  $2^n$  satisfying assignments.

In fact, counting the number of satisfying assignments to DNF formulas is  $\#P$ -complete.

## DNF Counting Algorithm I:

**Input:** A DNF formula  $F$  with  $n$  variables.

**Output:**  $Y =$  an approximation of  $c(F)$ .

- 1  $X = 0$ .
- 2 for  $k = 1$  to  $m$  do
  - 1 Generate a random assignment for the  $n$  variables, chosen uniformly at random from all  $2^n$  possible assignments.
  - 2 If the random assignment satisfies  $F$ , then  $X = X + 1$ .
- 3 return  $Y = (X/m)2^n$ .



## DNF Counting: A First Attempt

Let  $X_k$  be 1 if the  $k$ -th iteration in the algorithm generated a satisfying assignment and 0 otherwise. Then  $X = \sum_{k=1}^m X_k$  where the  $X_k$  are independent 0-1 random variables that each take the value 1 with probability  $c(F)/2^n$ . Hence, by linearity of expectations,

$$E[Y] = \frac{E[X]2^n}{m} = c(F).$$

It is not difficult to see that  $X/m$  gives an  $(\epsilon, \delta)$ -approximation of  $c(F)/2^n$ , and hence that  $Y$  gives an  $(\epsilon, \delta)$ -approximation of  $c(F)$ , when

$$m \geq \frac{2 \cdot 2^n \ln(2/\delta)}{\epsilon^2 c(F)}.$$

## Problem

*If  $c(F)$  is small, then the number of repetitions  $m$  is exponentially large, as*

$$m \geq \frac{2 \cdot 2^n \ln(2/\delta)}{\epsilon^2 c(F)}.$$

We now revise our sampling procedure to obtain an FPRAS.

Let

$$F = C_1 \vee C_2 \vee \cdots \vee C_t.$$

We omit clauses that include a variable and their negation.

If the clause  $C_k$  has  $\ell_k$  literals, then it is satisfied by

$$2^{n-\ell_k}$$

assignments.

Let  $S_k$  the set of assignments that satisfy clause  $C_k$ .

Let  $U = \{(k, v) \mid 1 \leq k \leq t, v \in S_k\}$

We know  $|U|$  and  $|S_k|$ , since

$$|U| = \sum_{k=1}^t |S_k|$$

and  $|S_k| = 2^{n-\ell_k}$ .

## Goal

We want to estimate

$$c(F) = \left| \bigcup_{k=1}^t S_k \right|$$

For this purpose, we investigate the following subset of  $U$ :

$$S = \{(k, v) \mid 1 \leq k \leq t, v \in S_k, v \notin S_j \text{ for } j < k\}.$$

In  $S$ , each assignment  $v$  occurs just once, so

$$|c(F)| = |S|.$$

We can estimate  $|S|$  by estimating the ratio  $|S|/|U|$ . We find this ratio by sampling from  $U$  uniformly at random. Since an assignment can occur in at most  $t$  sets  $S_k$ , we have

$$\frac{|S|}{|U|} \geq \frac{1}{t}.$$

So  $S$  is relatively dense in  $U$ .

How do we sample uniformly at random from  $U$ ?

Choose  $k$  with probability  $|S_k|/|U|$ . Then choose a satisfying assignment uniformly at random from  $S_k$ .

$$\begin{aligned}\Pr[(k, v) \text{ is chosen}] &= \Pr[k \text{ is chosen}] \Pr[v \text{ is chosen} \mid k \text{ is chosen}] \\ &= \frac{|S_k|}{|U|} \cdot \frac{1}{|S_k|} \\ &= \frac{1}{|U|}.\end{aligned}$$

## DNF Counting Algorithm II:

**Input:** A DNF formula  $F$  with  $n$  variables.

**Output:**  $Y =$  an approximation of  $c(F)$ .

- 1  $X = 0$ .
- 2 for  $k = 1$  to  $m$  do
  - 1 Choose  $i$  with probability  $|S_i|/|U|$ , and an assignment  $v$  from  $S_i$  uniformly at random.
  - 2 If  $v$  is not in any  $S_j$  for  $j < i$ , then  $X = X + 1$ .
- 3 return  $Y = (X/m)|U|$ .



### Proposition

*The DNF counting algorithm II is a fully polynomial randomized approximation scheme for the DNF counting problem when  $m = \lceil (3t/\epsilon^2) \ln(2/\delta) \rceil$ .*

The reason is that we choose pairs  $(i, v)$  uniformly at random from  $U$ . For the given number of repetitions  $m$ , the algorithm is an  $(\epsilon, \delta)$ -approximation to  $c(F) = |S|/|U|$  for each boolean function  $F$ .

# From Approximate Sampling to Approximate Counting

The example of DNF formulas shows that there is a fundamental connection between being able to **sample** from an appropriate space and being able to **count** the number of objects with some property in that space.

If you can sample the solutions to a so-called “self-reducible” combinatorial problem almost uniformly, then you can construct a randomized algorithm that approximately counts the number of solutions to that problem.

### Definition

Let  $w$  be the random output of a sampling algorithm for a finite sample space  $\Omega$ . We say that the sampling algorithm creates an  $\epsilon$ -**uniform sample** of  $\Omega$  if and only if

$$\left| \Pr[w \in S] - \frac{|S|}{|U|} \right| \leq \epsilon.$$

A sampling algorithm is a **fully polynomial almost uniform sampler** (or shortly FPAUS) if and only if given an input  $x$  and a parameter  $\epsilon > 0$ , it creates an  $\epsilon$ -uniform sample of  $\Omega(x)$  and runs in time polynomial in  $\ln(1/\epsilon)$  and the size of the input  $x$ .

## The Big Idea

FPRAS  $\iff$  FPAUS

# Markov Chain Monte Carlo Algorithms

## The Idea

Given a probability distribution  $\pi$  on a set  $S$ , we want to be able to sample from this probability distribution.

In MCMC, we define a Markov chain that has  $\pi$  as a stationary distribution. We run the chain for some iterations and then sample from it.

## The Idea

Given a probability distribution  $\pi$  on a set  $S$ , we want to be able to sample from this probability distribution.

In MCMC, we define a Markov chain that has  $\pi$  as a stationary distribution. We run the chain for some iterations and then sample from it.

## Why?

Sometimes it is easier to construct the Markov chain than the probability distribution  $\pi$ .



### Definition

Let  $G = (V, E)$  be a graph. The **hardcore model** of  $G$  randomly assigns either 0 or 1 to each vertex such that no neighboring vertices both have the value 1.

Assignment of the values 0 or 1 to the vertices are called **configurations**. So a configuration is a map in  $\{0, 1\}^V$ .

A configuration is called **feasible** if and only if no adjacent vertices have the value 1.

In the hardcore model, the feasible configurations are chosen uniformly at random.

### Question

For a given graph  $G$ , how can you directly choose a feasible configuration uniformly at random?

### Question

For a given graph  $G$ , how can you directly choose a feasible configuration uniformly at random?

An equivalent question is:

### Question

For a given graph  $G$ , how can you directly choose independent sets of  $G$  uniformly at random?

### Observation

In an  $n \times n$  grid graph, there are  $2^{n^2}$  configurations.

## Grid Graph Example

### Observation

In an  $n \times n$  grid graph, there are  $2^{n^2}$  configurations.

### Observation

There are at least  $2^{n^2/2}$  feasible configurations in the grid graph.

Indeed, set every other node in the grid graph to 0. For example, if we label the vertices by  $\{(x, y) \mid 0 \leq x < n, 0 \leq y < n\}$ . Then set all vertices with  $x + y \equiv 0 \pmod{2}$  to 0. The value of the remaining  $n^2/2$  vertices can be chosen arbitrarily, giving at least  $2^{n^2/2}$  feasible configurations.

Direct sampling from the feasible configurations seems difficult.

Given a graph  $G = (V, E)$  with a set  $\mathcal{F}$  of feasible configurations. We can define a Markov chain with state space  $\mathcal{F}$  and the following transitions

- 1 Let  $X_n$  be the current feasible configuration. Pick a vertex  $v \in V$  uniformly at random.
- 2 For all vertices  $w \in V \setminus \{v\}$ , the value of the configuration will not change:  $X_{n+1}(w) = X_n(w)$ .
- 3 Toss a fair coin. If the coin shows heads and all neighbors of  $v$  have the value 0, then  $X_{n+1}(v) = 1$ ; otherwise  $X_{n+1}(v) = 0$ .

### Proposition

*The hardcore model Markov chain is irreducible.*

### Proposition

*The hardcore model Markov chain is irreducible.*

### Proof.

Given an arbitrary feasible configuration with  $m$  ones, it is possible to reach the configuration with all zeros in  $m$  steps.

Similarly, it is possible to go from the zero configuration to an arbitrary feasible configuration with positive probability in a finite number of steps.

Therefore, it is possible to go from an arbitrary feasible configuration to another in a finite number of steps with positive probability. □



## Proposition

*The hardcore model Markov chain is aperiodic.*

### Proposition

*The hardcore model Markov chain is aperiodic.*

### Proof.

For each state, there is a small but nonzero probability that the Markov chain stays in the same state. Thus, each state is aperiodic. Therefore, the Markov chain is aperiodic. □

## Proposition

*Let  $\pi$  denote the uniform distribution on the set of feasible configurations  $\mathcal{F}$ . Let  $P$  denote the transition matrix. Then*

$$\pi_f P_{f,g} = \pi_g P_{g,f}$$

*for all feasible configurations  $f$  and  $g$ .*

## Proof.

Since  $\pi_f = \pi_g = 1/|\mathcal{F}|$ , it suffices to show that  $P_{f,g} = P_{g,f}$ .

- 1 This is trivial if  $f = g$ .
- 2 If  $f$  and  $g$  differ in more than one vertex, then  $P_{f,g} = 0 = P_{g,f}$ .
- 3 If  $f$  and  $g$  differ only on the vertex  $v$ . If  $G$  has  $k$  vertices, then

$$P_{f,g} = \frac{1}{2} \cdot \frac{1}{k} = P_{g,f}.$$



## Corollary

*The stationary distribution of the hardcore model Markov chain is the uniform distribution on the set of feasible configurations.*

# Uniform Distributions

## Problem

*If we define a Markov chain or a random walk, then the stationary distribution might not be uniform.*

*How can we obtain a Markov chain that has a stationary distribution that is uniform?*

## Idea

We are going to modify the transition probabilities and introduce self-loops so that the resulting stationary distribution is uniform.

## Definition

Let  $x$  be an element  $x$  of the state space  $S$  of a Markov chain with transition matrix  $P$ . We define the **neighborhood**  $N(x)$  of the state  $x$  as

$$N(x) = \{y \mid y \in S, P_{x,y} > 0\}.$$

In the graphical representation,  $N(x)$  are all vertices that can be reached from  $x$ .



## Proposition

*Suppose that we are given a random walk on a connected undirected graph with vertex set  $S$ . Let  $N$  denote the maximum number of neighbors of any state, so  $N = \max_{x \in S} |N(x)|$ . Let  $M$  be an integer such that  $M \geq N$ . Consider the Markov chain with state space  $S$  and transition matrix*

$$P_{x,y} = \begin{cases} 1/M & \text{if } x \neq y \text{ and } y \in N(x), \\ 0 & \text{if } x \neq y \text{ and } y \notin N(x), \\ 1 - |N(x)|/M & \text{if } x = y. \end{cases}$$

*The resulting Markov chain is irreducible and aperiodic. The stationary distribution of this chain is the uniform distribution on  $S$ .*

Proof.

Let  $\pi$  denote the uniform distribution on  $S$ , that is,

$$\pi_x = 1/|S|$$

for all  $x \in S$ .

If  $x$  and  $y$  are distinct adjacent elements of  $S$ , then

$$\pi_x P_{x,y} = \frac{1}{|S|} \cdot \frac{1}{M} = \pi_y P_{y,x}.$$

If  $x$  and  $y$  are distinct non-adjacent elements of  $S$ , then

$$\pi_x P_{x,y} = \frac{1}{|S|} \cdot 0 = \pi_y P_{y,x}.$$

Therefore,  $\pi$  is a reversible distribution for the Markov chain. The Markov chain is irreducible and aperiodic, so  $\pi$  is the stationary distribution.  $\square$